

Εισαγωγή στη C++

Ειδικά Θέματα Τηλεπικοινωνιών
Πρόγραμμα Μεταπτυχιακών Σπουδών Ραδιοηλεκτρολογίας

Αρχές προγραμματισμού

- Δομημένος προγραμματισμός
(structured programming)
- Αντικειμενοστρεφής προγραμματισμός
(object oriented programming)

Αρχές προγραμματισμού

Δομημένος προγραμματισμός

- ✓ Διαδικαστικές (procedural) γλώσσες προγραμματισμού
- ✓ Πρόγραμμα; → Λίστα εντολών (instructions)
- ✓ Μεγάλα προγράμματα; → Υποπρογράμματα:
 - συναρτήσεις (functions)
 - υπορουτίνες (subroutines)
 - διαδικασίες (procedures)
- ✓ Ομαδοποίηση υποπρογραμμάτων σε modules

Αρχές προγραμματισμού

Δομημένος προγραμματισμός

⇒ Πολυπλοκότητα!!!

Αρχές προγραμματισμού

Δομημένος προγραμματισμός

- ✓ Έμφαση στις ενέργειες, όχι στα δεδομένα
- ✓ Δεδομένα προσπελάσιμα από όλες τις συναρτήσεις
- ✓ Μεταβολή του είδους των δεδομένων; →
Αλλαγή όλων (!;) των συναρτήσεων
- ✓ Δημιουργία νέων τύπων δεδομένων;

Αρχές προγραμματισμού

Αντικειμενοστρεφής προγραμματισμός

- ✓ Συνδυασμός δεδομένων και συναρτήσεων που ενεργούν σε αυτά σε ένα αντικείμενο
- ✓ Συναρτήσεις μέλη του αντικειμένου
- ✓ Απόκρυψη και ενθυλάκωση των δεδομένων

Χαρακτηριστικά ΑΣ γλωσσών

- Αντικείμενα
- Τάξεις
- Κληρονομικότητα
- Επαναχρησιμοποίηση
- Δημιουργία νέων τύπων δεδομένων
- Πολυμορφισμός και υπέρβαση

Χαρακτηριστικά ΑΣ γλωσσών

Αντικείμενα

✓ Φυσικά αντικείμενα

Χαρακτηριστικά ΑΣ γλωσσών

Αντικείμενα

- ✓ Φυσικά αντικείμενα
- ✓ Στοιχεία του περιβάλλοντος χρήστη-Η/Υ

Χαρακτηριστικά ΑΣ γλωσσών

Αντικείμενα

- ✓ Φυσικά αντικείμενα
- ✓ Στοιχεία του περιβάλλοντος χρήστη-Η/Υ
- ✓ Κατασκευές προγραμματιστικές

Χαρακτηριστικά ΑΣ γλωσσών

Αντικείμενα

- ✓ Φυσικά αντικείμενα
- ✓ Στοιχεία του περιβάλλοντος χρήστη-Η/Υ
- ✓ Κατασκευές προγραμματιστικές
- ✓ Συλλογές δεδομένων

Χαρακτηριστικά ΑΣ γλωσσών

Αντικείμενα

- ✓ Φυσικά αντικείμενα
- ✓ Στοιχεία του περιβάλλοντος χρήστη-Η/Υ
- ✓ Κατασκευές προγραμματιστικές
- ✓ Συλλογές δεδομένων
- ✓ Τύποι δεδομένων ορισμένοι από το χρήστη

Χαρακτηριστικά ΑΣ γλωσσών

Τάξεις

- ✓ Συλλογές παρόμοιων αντικειμένων
- ✓ Κάθε αντικείμενο είναι μέλος μιας τάξης

Χαρακτηριστικά ΑΣ γλωσσών

Κληρονομικότητα

- ✓ Συλλογές παρόμοιων αντικειμένων
αρχική (base) και παράγωγες (derived) τάξεις
- ✓ Σχέση παράγωγης τάξης και αντικειμένου
- ✓ Ομαδοποίηση κοινών διαδικασιών

Χαρακτηριστικά ΑΣ γλωσσών

Επαναχρησιμοποίηση

- ✓ Χρήση υπάρχουσας τάξης σε άλλο πρόγραμμα
- ✓ +κληρονομικότητα → ευελιξία

Χαρακτηριστικά ΑΣ γλωσσών

Πολυμορφισμός και υπέρβαση

- ✓ Ενεργούσα συνάρτηση ανάλογη των δεδομένων
- ✓ Τελεστές υπέρβασης (overloading)

Δημιουργία εκτελέσιμου προγράμματος

- Συγγραφή πηγαίου κώδικα
(source code)
- Μεταγλώττιση σε αντικειμενικό κώδικα
(object code)
- Σύνδεση (linking) σε εκτελέσιμο κώδικα
(executable code)

Δημιουργία εκτελέσιμου προγράμματος

```
// my first program  
  
#include <iostream.h>  
  
void main( )  
    {  
        cout << "...";  
    }
```

Αρχές προγραμματισμού με τη C++

Δομή προγράμματος

- ✓ Συναρτήσεις
- ✓ Προτάσεις προγράμματος
- ✓ Λευκά διαστήματα
- ✓ Οδηγίες προεπεξεργαστή
- ✓ Αρχεία κεφαλίδων (header files)
- ✓ Σχόλια

Δομή προγράμματος της C++

οδηγίες
προεπεξεργαστή

```
// my first program
```

σχόλιο

αρχεία κεφαλίδων

```
#include <iostream.h>
```

λευκά διαστήματα

συνάρτηση

```
void main(
```

```
{
```

```
cout << "...";
```

```
}
```

πρόταση

Αρχές προγραμματισμού με τη C++

Μεταβλητές

- ✓ Τι είναι;
- ✓ Ονοματολογία
- ✓ Τύποι μεταβλητών
- ✓ Δήλωση και ορισμός μεταβλητής
- ✓ Αρχική απόδοση τιμής
- ✓ Παραστάσεις μεταβλητών

Αρχές προγραμματισμού με τη C++

Τύποι μεταβλητών

✓ Ακέραιες

Αρχές προγραμματισμού με τη C++

Τύποι μεταβλητών

- ✓ Ακέραιες
- ✓ Μεταβλητές χαρακτήρων

Αρχές προγραμματισμού με τη C++

Τύποι μεταβλητών

✓ **Ακέραιες**

✓ **Μεταβλητές χαρακτήρων**

ακολουθίες διαφυγής (escape sequences):

<code>\a</code>	bell
<code>\b</code>	backspace
<code>\f</code>	feed
<code>\n</code>	new line
<code>\r</code>	carriage return
<code>\t</code>	tab
<code>\\</code>	backslash
<code>\'</code>	single quotation
<code>\"</code>	double quotation
<code>\xdd</code>	hexadecimal representation

Αρχές προγραμματισμού με τη C++

Τύποι μεταβλητών

- ✓ Ακέραιες
- ✓ Μεταβλητές χαρακτήρων
- ✓ Κινητής υποδιαστολής
- ✓ Λογικού τύπου (boolean)

Αρχές προγραμματισμού με τη C++

Τύποι μεταβλητών (32bit)

<u>Type</u>	<u>Size (bits)</u>	<u>Range</u>	<u>Sample applications</u>
unsigned char	8	$0 \leq X \leq 255$	Small numbers and full PC character set
char	8	$-128 \leq X \leq 127$	Very small numbers and ASCII characters
short int	16	$-32,768 \leq X \leq 32,767$	Counting, small numbers, loop control
unsigned int	32	$0 \leq X \leq 4,294,967,295$	Large numbers and loops
int	32	$-2,147,483,648 \leq X \leq 2,147,483,647$	Counting, small numbers, loop control
unsigned long	32	$0 \leq X \leq 4,294,967,295$	Astronomical distances
enum	32	$-2,147,483,648 \leq X \leq 2,147,483,647$	Ordered sets of values
long	32	$-2,147,483,648 \leq X \leq 2,147,483,647$	Large numbers, populations
float	32	$1.18 \cdot 10^{-38} < X < 3.40 \cdot 10^{38}$	Scientific (7-digit) precision
double	64	$2.23 \cdot 10^{-308} < X < 1.79 \cdot 10^{308}$	Scientific (15-digit) precision
long double	80	$3.37 \cdot 10^{-4932} < X < 1.18 \cdot 10^{4932}$	Financial (18-digit) precision

Αρχές προγραμματισμού με τη C++

Είσοδος - Έξοδος

✓ Τελεστής εισαγωγής

Αρχές προγραμματισμού με τη C++

Είσοδος - Έξοδος

- ✓ Τελεστής εισαγωγής
- ✓ Τελεστής εξαγωγής

Αρχές προγραμματισμού με τη C++

Είσοδος - Έξοδος

- ✓ Τελεστής εισαγωγής
- ✓ Τελεστής εξαγωγής
- ✓ Μορφοποίηση εξόδου

Είσοδος - Έξοδος

```
// calculation of circle area
```

```
#include <iostream.h>
```

```
void main( )
```

```
{
```

```
float radius, area;
```

```
const float PI=3.14159;
```

```
cout << "Please, give the circle radius: ";
```

```
cin >> radius;
```

```
area=PI*radius*radius;
```

```
cout << "The area is " << area;
```

```
}
```

Είσοδος - Έξοδος

```
// calculation of circle area

#include <iostream.h>
#define PI 3.14159

void main( )
{
    float radius, area;

    cout << "Please, give the circle radius: ";
    cin >> radius;
    area=PI*radius*radius;
    cout << "The area is " << area;
}
```

Μορφοποίηση εισόδου - εξόδου

```
// calculation of circle characteristics

#include <iostream.h>
#define PI 3.14159

void main( )
{
    float radius,area,circum;

    cout << "Please, give the circle radius: ";
    cin >> radius;
    cout << "The circle with radius " << radius << endl;
    area=PI*radius*radius;
    cout << "has an area of " << area << endl;
    circum=2*PI*radius;
    cout << "and a circumference of " << circum;
}
```


Μορφοποίηση εισόδου - εξόδου

```
// calculation of circle characteristics

#include <iostream.h>
#include <iomanip.h>
#define PI 3.14159

void main( )
{
    float radius,area,circum;

    cout << "Please, give the circle radius: ";
    cin >> radius;
    area=PI*radius*radius;
    circum=2*PI*radius;
    cout << left << setw(24) << "The circle with radius "
         << fixed << right << setw(12) << setprecision(3) << radius << endl
         << left << setw(24) << "has an area of "
         << fixed << right << setw(12) << setprecision(3) << area << endl
         << left << setw(24) << "and a circumference of "
         << fixed << right << setw(12) << setprecision(3) << circum << endl;
}
```

Αρχές προγραμματισμού με τη C++

Μετατροπή τύπου μεταβλητής

- ✓ Αυτόματη μετατροπή
- ✓ Προσαρμογή

Μετατροπή τύπου μεταβλητής

```
#include <iostream.h>

void main( )
{
    int var1=1000000, var2;

    var2=(var1*1000000)/1000000;
    cout << "var2 = " << var2 << endl;
}
```

Μετατροπή τύπου μεταβλητής

```
#include <iostream.h>

void main( )
{
    int var1=1000000, var2;

    var2=(float(var1)*1000000)/1000000;
    cout << "var2 = " << var2 << endl;
}
```

Αρχές προγραμματισμού με τη C++

Βασικοί τελεστές

- ✓ Τελεστής απόδοσης τιμής
- ✓ Τελεστής υπολοίπου
- ✓ Αριθμητικοί τελεστές
- ✓ Τελεστές αύξησης και μείωσης
- ✓ Αρχείο κεφαλίδας <math.h>
- ✓ Προτεραιότητα τελεστών

Βρόχοι και Αποφάσεις

Οι **προτάσεις ελέγχου** περιέχουν

λογικές παραστάσεις που περιλαμβάνουν

συσχετιστικούς τελεστές και είναι

αληθείς ή ψευδείς

Βρόχος for

```
for(αρχική τιμή; παράσταση ελέγχου; παράσταση μεταβολής)
{
    σώμα βρόχου
}
```

Παράδειγμα:

```
for (int i=0; i<100; i++)
{
    cout << i << endl;
}
```

Βρόχος while

```
while(παράσταση ελέγχου)
{
    σώμα βρόχου
}
```

Παράδειγμα:

```
int i=0;
while (i<100) {
    cout << i << endl;
    i++;
}
```


Βρόχος do-while

```
do {  
    σώμα βρόχου  
}  
while(παράσταση ελέγχου);
```

Παράδειγμα:

```
int i=0;  
do    {  
    cout << i << endl;  
    i++;  
    }  
while (i<100) ;
```

Απόφαση με την πρόταση if

```
if (παράσταση ελέγχου)
{
    σώμα if πρότασης
}
```

Παράδειγμα:

```
for(int i=0;i<10;i++) {
    if ((i % 3) == 0)
        cout << i << " is a multiple of 3" << endl;
}
```

Απόφαση με την πρόταση if-else

```
if (παράσταση ελέγχου) {  
    σώμα if πρότασης  
}  
else {  
    σώμα else πρότασης  
}
```

Παράδειγμα:

```
for(int i=0;i<10;i++) {  
    if ((i % 3) == 0)  
        cout << i << " is a multiple  
of 3" << endl;  
        else  
            not a multiple of 3" << endl;  
}
```

Απόφαση με την πρόταση switch

```
switch (μεταβλητή ακέραιη ή χαρακτήρα) {  
    case 1:  
        πρόταση;  
        .....  
        break;  
    case 2:  
        πρόταση;  
        .....  
        break;  
    default:  
        πρόταση;  
        .....  
        πρόταση;  
}
```

Απόφαση με την πρόταση switch

Παράδειγμα:

```
switch (c) {
    case 'y':
        cout << "you have pressed: \'y\'" << endl;
        break;
    case 'Y':
        cout << "you have pressed: \'Y\'" << endl;
        break;
    default:
        cout << c << " is not a valid choice!" << endl;
}
```

Απόφαση με παράσταση υπό συνθήκη

αποτέλεσμα=(παράσταση ελέγχου)?παράσταση 1:παράσταση 2;

Παράδειγμα:

```
min = (alpha<beta)? Alpha: beta;
```

Οι προτάσεις break και continue

- Η πρόταση **break** προκαλεί έξοδο από ένα βρόχο
- Η πρόταση **continue** μεταφέρει τη ροή του προγράμματος στην αρχή ενός βρόχου

Δομές (Structures)

Προσδιορισμός μιας δομής

```
struct όνομα δομής  
{  
    μέλη δομής  
};
```

Παράδειγμα:

```
struct Course  
{  
    int CourseCode;  
    int Semester;  
    int Instructor;  
};
```


Δομές (Structures)

Ορισμός μεταβλητής τύπου δομής

Παράδειγμα:

...

```
Course Telecommunications;
```

...

Προσπέλαση μελών δομής

Παράδειγμα:

...

```
Telecommunications.Semester=4;
```

...

Δομές (Structures)

Συνδυασμός προσδιορισμού και ορισμού

Παράδειγμα:

```
struct  
{  
int CourseCode;  
int Semester;  
int Instructor;  
} Telecommunications;
```

Αρχικοποίηση δομής

Παράδειγμα:

```
...  
Course Telecommunications={353,4,12};  
...
```

Δομές (Structures)

Ένθετες δομές

Παράδειγμα:

```
struct Point
{
int x;
int y;
};

struct StraightLine
{
Point StartPoint;
Point EndPoint;
};
```

Δομές (Structures)

Προσπέλαση ένθετων δομών

Παράδειγμα:

```
...  
MyStraightLine.StartPoint.x=0;  
MyStraightLine.StartPoint.y=0;  
...
```

Αρχικοποίηση ένθετων δομών

Παράδειγμα:

```
...  
StraightLine MyStraightLine={{0,0},{1,1}};  
...
```

Δομές (Structures)

- Τελεστές; \Rightarrow Πρόβλημα
- Δομές και τάξεις αντικειμένων

Απαριθμητοί (enumerated) τύποι δεδομένων

Σύνταξη

enum όνομα μεταβλητής { λίστα τιμών-μελών };

Παράδειγμα:

```
enum DayOfTheWeek { Sun, Mon, Tue, Wed, Thu, Fri, Sat } ;  
...  
DayOfTheWeek day1, day2 ;  
...
```

Απαριθμητοί (enumerated) τύποι δεδομένων

- Μεταβλητές ισοδύναμες με ακεραίους
- Καθορισμός αρχικής τιμής

Παράδειγμα:

```
enum Position {low=-1,middle,high};
```

Συναρτήσεις

Δήλωση συνάρτησης

Παράδειγμα:

```
void anyFunction(int, int );
```

Κλήση συνάρτησης

Παράδειγμα:

```
anyFunction(10, a);
```


Συναρτήσεις

Ορισμός συνάρτησης

```
τύπος-επιστρεφ.-τιμής όνομα-συνάρτησης (ορίσματα)  
{  
  ...  
  προτάσεις;  
  ...  
}
```

Συναρτήσεις

Ορισμός συνάρτησης

Παράδειγμα:

```
void RepeatChar(char ch, int n)
{
    for(int j=0; j<n; j++)
        cout << ch;
    cout << endl;
}
```

Συναρτήσεις

- Μεταβίβαση μεταβλητών με τιμή
- Μεταβίβαση δομών με τιμή

Παράδειγμα:

```
void DisplayCourse (Course AnyCourse)
{
    cout << "The code of the course is "
         << AnyCourse.CourseCode << endl;
    ...
}
```

Συναρτήσεις

- Επιστροφή τιμών
- Η πρόταση return

Παράδειγμα:

```
float InchToCentimetre(float InchVal)
{
    float CmVal=InchVal*2.54;
    return CmVal;
}
```

Συναρτήσεις

Επιστροφή μεταβλητών δομής

Παράδειγμα:

```
Course InputCourse ()
{
    Course NewCourse;
    cin >> NewCourse.Code
        >> NewCourse.Semester
        >> NewCourse.Instructor;
    return NewCourse;
}
```

Συναρτήσεις

Μεταβίβαση ορισμάτων με αναφορά

Παράδειγμα:

```
void Swap(float& Val1, float& val2)
{
    float temp;
    temp=Val2;
    Val2=Val1;
    Val1=temp;
}
```

Συναρτήσεις

Συναρτήσεις με υπέρβαση

Διαφορετικός αριθμός ορισμάτων

Παράδειγμα:

```
float logarithm(float Number)
{
    return log(Number) ;
}
```

```
float logarithm(float Number, int Base)
{
    return log(Number) / log(Base) ;
}
```

Συναρτήσεις

Συναρτήσεις με υπέρβαση

Διαφορετικά είδη ορισμάτων

Παράδειγμα:

```
float square(int Number)
{
    return Number*Number;
}
```

```
float square(float Number)
{
    return Number*Number;
}
```


Συναρτήσεις

Προεπιλεγμένα ορίσματα

Παράδειγμα:

```
float power(float, int=2);  
...  
float power(float Number, int Exp)  
{  
    float Result=1;  
    for (int i=0;i<Exp;i++)  
        Result=Result*Number;  
    return Result;  
}
```

Μεταβλητές

- Αυτόματες (τοπικές) μεταβλητές
- Εξωτερικές (καθολικές) μεταβλητές
- Στατικές μεταβλητές

Στατικές αυτόματες

Στατικές εξωτερικές

Πίνακες

- Ομαδοποίηση στοιχείων ίδιου τύπου
- Προσπέλαση στοιχείων με δείκτη
- Ο τύπος είτε βασικός είτε από τον χρήστη

Πίνακες

Ορισμός πίνακα

τύπος-δεδομένων όνομα-πίνακα[μέγεθος-πίνακα];

Παράδειγμα:

```
int Numbers[4];
```

Πίνακες

Μονοδιάστατος πίνακας στη μνήμη του Η/Υ

0xDEF8	25	Numbers [0]
0xDEF9	3	Numbers [1]
0xDEFA	17	Numbers [2]
0xDEFB	500	Numbers [3]

Πίνακες

Προσπέλαση στοιχείων πίνακα

Παράδειγμα:

```
...  
int Numbers[4];  
...  
Numbers[0]=25;  
...  
cin >> Numbers[1];  
...  
cout Numbers[3];  
...
```

Πίνακες

Αρχικοποίηση πίνακα (1/2)

Παράδειγμα:

```
...  
int Numbers[4]={1,2,3};  
...
```

Πίνακες

Αρχικοποίηση πίνακα (2/2)

Παράδειγμα:

```
...  
int Numbers[]={1,2,3};  
...
```


Πίνακες

Πολυδιάστατοι πίνακες

Παράδειγμα:

```
...  
int Numbers[Nx][Ny];  
...  
Numbers[2][4]=123;  
...  
cin >> Numbers[i][j];  
...  
cout Numbers[12][j];  
...
```

Πίνακες

Δισδιάστατος πίνακας στη μνήμη του Η/Υ

0xDEF8	25	Numbers[0][0]
0xDEF9	3	Numbers[0][1]
0xDEFA	17	Numbers[0][2]
0xDEFB	500	Numbers[1][0]
0xDEFC	12	Numbers[1][1]
0xDEFD	894	Numbers[1][2]

Πίνακες

Αρχικοποίηση πολυδιάστατου πίνακα

Παράδειγμα:

```
...  
int  
Numbers [2] [3] = { {1, 25, 37}, {12, 15, 18} };  
...
```

Πίνακες και συναρτήσεις

```
#include <iostream.h>

const int TotalFloats=10; // number of the float numbers!

float meanNumber(float FloatNumbers[TotalFloats]);

void main( )
{
    float Numbers[TotalFloats];
    cout << "Give the numbers: " << endl;
    for (int i=0;i<TotalFloats;i++) {
        cout << "Number " << (i+1) << " is ";
        cin >> Numbers[i];
    }
    cout << "The mean is " << meanNumber(Numbers) << endl;
}

float meanNumber(float FloatNumbers[TotalFloats])
{
    float TotalSum=0.0;
    for (int i=0;i<TotalFloats;i++)
        TotalSum+=FloatNumbers[i];
    return TotalSum/TotalFloats;
}
```

Πίνακες

Προσοχή!!!

Η μεταβίβαση πίνακα σε συνάρτηση γίνεται πάντα με αναφορά

Πίνακες δομών

```
#include <iostream.h>

const int TotalCourses=3; // number of the float numbers!

struct course {
    int CourseCode;
    int Semester;
};

course InputCourse();

void main( )
{
    course AllCourses[TotalCourses];
    cout << "Input the courses: " << endl;
    for (int i=0;i<TotalCourses;i++) {
        cout << "Course " << (i+1) << ":" << endl;
        AllCourses[i]=InputCourse();
    }
    cout << AllCourses[2].CourseCode << endl;
}

course InputCourse()
{
    course AnyCourse;
    cout << "Course code : ";
    cin >> AnyCourse.CourseCode;
    ...
    return AnyCourse;
}
```

Αλφαριθμητικά

Αλφαριθμητικές μεταβλητές

`char όνομα-μεταβλητής[μέγεθος+1];`

Παράδειγμα:

```
char str[120];
```

Αλφαριθμητικά

Αλφαριθμητικές στη μνήμη του Η/Υ

0xDEF8

G

str[0]

0xDEF9

o

str[1]

0xDEFa

d

str[2]

0xDEFB

\0

str[3]

Αλφαριθμητικά

Εισαγωγή και εμφάνιση αλφαριθμητικών

Παραδείγματα:

```
...  
char str[120];  
...  
cin >> str;  
...  
cin >> setw(120) >> str;  
...  
cout << str;
```

Αλφαριθμητικά

Αλφαριθμητικές σταθερές

Παραδείγματα:

```
char str[]="hello";  
char str[]={ 'h' , 'e' , 'l' , 'l' , 'o' };
```

Αλφαριθμητικά

Εισαγωγή με τη συνάρτηση `cin::get()`

Παραδείγματα:

```
...  
char str[100];  
...  
cin.get(str, 100);  
...  
cin.get(str, 100, ' %' );
```

Αντιγραφή αλφαριθμητικού

```
#include <iostream.h>
#include <string.h>

const int StringMax=100;

void main( )
{
    char String1[]="This is the string";
    char String2[StringMax];
    int i;

    for(i=0;i<strlen(String1);i++)
        String2[i]=String1[i];
    String2[i]='\0';
    cout << endl << String2 << endl;
}
```

Πίνακες αλφαριθμητικών

```
#include <iostream.h>

const int StringMax=100;
const int MaxNames=6;

void main( )
{
    char Names[MaxNames][StringMax]=
        {"akis", "makis", "takis", "vakis", "sakis", "lakis"};
    int i;

    cout << "The Names are:" << endl;
    for(i=0; i<MaxNames; i++)
        cout << Names[i] << endl;
}
```

Τάξεις

Ορισμός τάξης

```
class όνομα-τάξης  
{  
  
    private:  
    ...  
    protected:  
    ...  
    public:  
    ...  
};
```

Τάξεις

Ορισμός τάξης

Παράδειγμα:

```
class Simple
{
    private:
        int data;
    public:
        void setdata(int d) {data=d;}
        void showdata() {cout << data << endl;}
};
```

Τάξεις

Ορισμός τάξης (1/2)

Παράδειγμα αρχείου κεφαλίδας (“Simple.h”):

```
class Simple
{
private:
    int data;
public:
    void setdata(int );
    void showdata();
};
```


Τάξεις

Ορισμός τάξης (2/2)

Παράδειγμα πηγαίου κώδικα:

```
#include "Simple.h"

void Simple::setdata(int d)
{
    data=d;
}

void Simple::showdata()
{
    cout << data << endl;
}
```

Αντικείμενα

Μέθοδος εγκατάστασης (1/2)

Παράδειγμα αρχείου κεφαλίδας ("Counter.h"):

```
#include <iostream.h>
class Counter
{
    private:
        unsigned int count;
    public:
        Counter( )          { count = 0; }
        void inc_count( )  { count++; }
        int get_count( )   { return count; }
};
```

Αντικείμενα

Μέθοδος εγκατάστασης (2/2)

Παράδειγμα πηγαίου κώδικα:

```
#include "Counter.h"
void main( )
{
    Counter c1, c2;
    cout << "\nc1=" << c1.get_count( );
    cout << "\nc2=" << c2.get_count( );
    c1.inc_count( );
    c2.inc_count( );
    c2.inc_count( );
    cout << "\nc1=" << c1.get_count( );
    cout << "\nc2=" << c2.get_count( );
}
```

Αντικείμενα

Μέθοδος αποσύνδεσης

Παράδειγμα αρχείου κεφαλίδας (“Counter.h”):

```
#include <iostream.h>
class Counter
{
    private:
        unsigned int count;
    public:
        Counter( )          { count = 0; }
        ...
        ~Counter() { }
};
```

Αντικείμενα ως ορίσματα (1/3)

Παράδειγμα αρχείου κεφαλίδας ("Distance.h"):

```
class Distance
{
    private:
        int feet;
        float inches;
    public:
        Distance( ) { }
        Distance(int ft, float in);
        void getdist( );
        void showdist( );
        void add_dist( Distance, Distance );
};
```

Αντικείμενα ως ορίσματα (2/3)

Παράδειγμα πηγαίου κώδικα ("Distance.cpp"):

```
#include "Distance.h"
Distance::Distance(int ft, float in) {
    feet = ft; inches = in; }

void Distance::getdist( ) {
    cout << "\nGive feet: "; cin >> feet;
    cout << "Give inches: "; cin >> inches;
}

void Distance::showdist( ) {
    cout << feet << "'-" << inches << "'";
}

void Distance::add_dist(Distance d2, Distance d3) {
    inches = d2.inches + d3.inches;
    feet = 0;
    if(inches >= 12.0){
        inches -= 12.0;
        feet++; }
    feet += d2.feet + d3.feet;
}
```

Αντικείμενα ως ορίσματα (3/3)

Παράδειγμα πηγαίου κώδικα ("MainProgram.cpp") :

```
#include "Distance.h"
void main( )
{
    Distance dist1, dist3;
    Distance dist2(11, 6.25);
    dist1.getdist( );
    dist3.add_dist(dist1, dist2);
    cout << "\ndist1 = "; dist1.showdist( );
    cout << "\ndist2 = "; dist2.showdist( );
    cout << "\ndist3 = "; dist3.showdist( );
}
```

Επιστροφή αντικειμένων (1/3)

Παράδειγμα αρχείου κεφαλίδας ("Distance.h"):

```
class Distance
{
    private:
        int feet;
        float inches;
    public:
        Distance( ) { }
        Distance(int ft, float in);
        void getdist( );
        void showdist( );
        Distance add_dist( Distance );
};
```


Επιστροφή αντικειμένων (2/3)

Παράδειγμα πηγαίου κώδικα ("Distance.cpp"):

```
#include "Distance.h"
Distance::Distance(int ft, float in) {
    feet = ft; inches = in; }

...

Distance Distance::add_dist(Distance d2) {
    Distance temp;
    temp.inches=inches + d2.inches;
    if(temp.inches >= 12.0){
        temp.inches -= 12.0;
        temp.feet=1; }
    temp.feet += feet + d2.feet;
    return temp;
}
```

Επιστροφή αντικειμένων (3/3)

Παράδειγμα πηγαίου κώδικα ("MainProgram.cpp") :

```
#include "Distance.h"
void main( )
{
    Distance dist1, dist3;
    Distance dist2(11, 6.25);
    dist1.getdist( );
    dist3=dist1.add_dist(dist2);
    cout << "\ndist1 = "; dist1.showdist( );
    cout << "\ndist2 = "; dist2.showdist( );
    cout << "\ndist3 = "; dist3.showdist( );
}
```

Τάξεις και Αντικείμενα στη Μνήμη

- Μεταβλητές περίπτωσης - ομότυπα
- Αποθήκευση δεδομένων - μεθόδων
- Στατικά δεδομένα τάξης

Τάξεις και Αντικείμενα στη Μνήμη

Στατικά δεδομένα τάξης

Παράδειγμα:

```
#include <iostream.h>
class foo {
    private:
        static int count;
    public:
        foo( ) { count++; }
        int getcount( ) { return count; }
};
void main( )
{
    foo f1, f2, f3;
    cout << "\ncount is: " << f1.getcount( );
    cout << "\ncount is: " << f2.getcount( );
    cout << "\ncount is: " << f3.getcount( );
}
```

Υπέρβαση τελεστών

Υπέρβαση μοναδιαίων τελεστών (1/2)

Παράδειγμα:

```
#include <iostream.h>
class Counter
{
private:
    unsigned int count;
public:
    Counter( )      { count = 0; }
    int get_count( ) { return count; }
    void operator ++ ( ) { count++; }
};
void main( )
{
    Counter c2;
    cout << "\nc2=" << c2.get_count( );
    c2++;
    ++c2;
    cout << "\nc2=" << c2.get_count( );
}
```

Υπέρβαση τελεστών

Υπέρβαση μοναδιαίων τελεστών (2/2)

Παράδειγμα:

```
#include <iostream.h>
class Counter
{
    ...
    Counter operator ++ ( ) {
        count++;
        Counter temp;
        Counter temp.count = count;
        return temp;
    }
};
void main( ) {
    Counter c1,c2;
    cout << "\nc2=" << c2.get_count( );
    c2++;
    c1=c2++;
    cout << "\nc2=" << c2++.get_count( );
}
```

Υπέρβαση δυαδικών τελεστών (1/3)

“Distance.h”

```
#include <iostream.h>
class Distance
{
    private:
        int feet;
        float inches;
    public:
        Distance( ) { feet = 0; inches = 0.0; }
        Distance(int ft, float in) { feet = ft; inches = in; }
        void getdist( ) {
            cout << "give feet: "; cin >> feet;
            cout << "give inches: "; cin >> inches;
        }
        void showdist( ) { cout << feet << "'-" << inches << "'"; }
        Distance operator + ( Distance );
};
```

Υπέρβαση δυαδικών τελεστών (2/3)

“Distance.cpp”

```
#include "Distance.h"
Distance Distance::operator + (Distance d2)
{
    int f = feet + d2.feet;
    float i = inches + d2.inches;
    if(i >= 12.0)
    {
        i -= 12.0;
        f++;
    }
    return Distance(f,i);
}
```


Υπέρβαση δυαδικών τελεστών (3/3)

“MainProgram.cpp”

```
#include "Distance.h"
void main( )
{
    Distance dist1, dist3, dist4;
    dist1.getdist( );
    Distance dist2(11, 6.25);
    dist3 = dist1 + dist2;
    dist4 = dist1 + dist2 + dist3;
    cout << "\ndist1 = "; dist1.showdist( );
    cout << "\ndist2 = "; dist2.showdist( );
    cout << "\ndist3 = "; dist3.showdist( );
    cout << "\ndist4 = "; dist4.showdist( );
}
```

Μετατροπή δεδομένων

αντικείμενο προορισμού



`objectA = objectB`



αντικείμενο προέλευσης

Μετατροπή δεδομένων

Μετατροπές Τύπων

	<u>αντικείμενο προορισμού</u>	<u>αντικείμενο προέλευσης</u>
βασικός σε βασικό	-	-
βασικός σε τάξη	constructor	-
τάξη σε βασικό	-	μέθοδος μετατροπής
τάξη σε τάξη	constructor	μέθοδος μετατροπής

Κληρονομικότητα

- Διαδικασία δημιουργίας παραγώγων τάξεων (derived classes) από βασικές τάξεις (base classes)
- Επαναχρησιμοποίηση κώδικα

Κληρονομικότητα

Σύνταξη

Παράδειγμα:

```
#include "Counter.h"
class CounterDown : public Counter
{
public:
    Counter operator-- () {
        count--;
        return Counter(count) ;
    } ;
```

Κληρονομικότητα

- Προσπέλαση μελών βασικής τάξης:

Μέθοδος εγκατάστασης
Μέθοδοι βασικής τάξης

- Καθοριστές πρόσβασης:

καθοριστής	τάξη	παράγωγη τάξη	αντικείμενο εκτός τάξης
public	+	+	+
protected	+	+	-
private	+	-	-

Κληρονομικότητα

Μέθοδος εγκατάστασης παράγωγης τάξης

Παράδειγμα:

```
#include "Counter.h"
class CountDn : public Counter
{
    public:
        CountDn( ) : Counter( ) { }
        CountDn(int c) : Counter(c) { }
        CountDn operator -- ( ) {
            count--;
            return CountDn(count);
        }
};
```

Κληρονομικότητα

- Υπέρβαση μεθόδων βασικής τάξης
- Διάκριση μεταξύ overloaded μεθόδων
- Διάκριση με τον τελεστή '::'

Δημόσια και ιδιωτική κληρονομικότητα(1/2)

```
class A
{
private:
    int privdataA;
protected:
    int protdataA;
public:
    int pubdataA;
};
class B : public A
{
public:
    void funct( )
    {
        int a;
        a = privdataA;
        a = protdataA;
        a = pubdataA;
    }
};
class C : private A
{
public:
    void funct( )
    {
        int a;
        a = privdataA;
        a = protdataA;
        a = pubdataA;
    }
};
```

Δημόσια και ιδιωτική κληρονομικότητα (2/2)

```
void main( )
    int a;
    B objB;
    a = objB.privdataA;
    a = objB.protdataA;
    a = objB.pubdataA;
    C objC;
    a = objC.privdataA;
    a = objC.protdataA;
    a = objC.pubdataA;
}
```

Δείκτες

- Σειριακή μνήμη Η/Υ
- Μονοσήμαντη - μονότονη αρίθμηση
- Μέγεθος μεταβλητών

ΔΕΙΚΤΕΣ

Τύποι μεταβλητών (32bit)

<u>Type</u>	<u>Size (bits)</u>	<u>Range</u>	<u>Sample applications</u>
unsigned char	8	$0 \leq X \leq 255$	Small numbers and full PC character set
char	8	$-128 \leq X \leq 127$	Very small numbers and ASCII characters
short int	16	$-32,768 \leq X \leq 32,767$	Counting, small numbers, loop control
unsigned int	32	$0 \leq X \leq 4,294,967,295$	Large numbers and loops
int	32	$-2,147,483,648 \leq X \leq 2,147,483,647$	Counting, small numbers, loop control
unsigned long	32	$0 \leq X \leq 4,294,967,295$	Astronomical distances
enum	32	$-2,147,483,648 \leq X \leq 2,147,483,647$	Ordered sets of values
long	32	$-2,147,483,648 \leq X \leq 2,147,483,647$	Large numbers, populations
float	32	$1.18 \cdot 10^{-38} < X < 3.40 \cdot 10^{38}$	Scientific (7-digit) precision
double	64	$2.23 \cdot 10^{-308} < X < 1.79 \cdot 10^{308}$	Scientific (15-digit) precision
long double	80	$3.37 \cdot 10^{-4932} < X < 1.18 \cdot 10^{4932}$	Financial (18-digit) precision

ΔΕΪΚΤΕΣ

Τελεστής διεύθυνσης &

Παράδειγμα:

```
#include <iostream.h>
void main()
{
    int var=1;
    double fvar=1.0;

    cout << &var << endl
         << &fvar << endl;
}
```

ΔΕΪΚΤΕΣ

Μεταβλητές δείκτη

Παράδειγμα:

```
#include <iostream.h>
void main()
{
    int var=1;
    int* p_var;
    double fvar=1.0;
    double *p_fvar;
    p_var=&var;
    p_fvar=&fvar;
    cout << p_var << endl
         << p_fvar << endl;
}
```

ΔΕΪΚΤΕΣ

Προσπέλαση μεταβλητής που δείχνει ο δείκτης Τελεστής έμμεσης αναφοράς * (1/2)

Παράδειγμα:

```
#include <iostream.h>
void main()
{
    int var=1;
    int* p_var;
    p_var=&var;

    cout << "variable in address: " << p_var
         << " has the value: " << *p_var
         << endl;
}
```

ΔΕΙΚΤΕΣ

Προσπέλαση μεταβλητής που δείχνει ο δείκτης Τελεστής έμμεσης αναφοράς * (2/2)

Παράδειγμα:

```
#include <iostream.h>
void main()
{
    int var1,var2;
    int* p_var;
    p_var=&var1; // δώσε στο δείκτη p_var τη
                // διεύθυνση της μεταβλητής var1
    *p_var=1;    // δώσε στη μεταβλητή στην οποία δείχνει ο
                // δείκτης p_var (δηλ. στη μεταβλητή var1
                // την τιμη 1
    var2=*p_var; // ισοδύναμο με var2=1;

    cout << "var1=" << var1 << endl << "var2=" << var2 << endl;
}
```


ΔΕΙΚΤΕΣ

Δείκτης για τύπο void

Παράδειγμα:

```
#include <iostream.h>
void main()
{
    int var=1;
    int* p_var;
    double fvar=1.0;
    void* p_void;

    p_var=&fvar; // λάθος
    p_void=&fvar; // σωστό
}
```

ΔΕΪΚΤΕΣ

ΔΕΪΚΤΕΣ για πίνακες (1/2)

Παράδειγμα:

```
#include <iostream.h>
void main()
{
    int i,iarray={ 2, 45, 39 };
    for(i=0;i<3;i++)
        cout << iarray[i] << endl;
    for(i=0;i<3;i++)
        cout << *(iarray+i) << endl;
    for(i=0;i<3;i++)
        cout << *(iarray++) << endl; // λάθος
}
```

ΔΕΪΚΤΕΣ

ΔΕΪΚΤΕΣ γΙΑ ΠΪΝΑΚΕΣ (2/2)

Παράδειγμα:

```
#include <iostream.h>
void main()
{
    int i,iarray={ 2, 45, 39 };
    int *parray;
    parray=iarray;

    for(i=0;i<3;i++)
        cout << *(parray+i) << endl;
    for(i=0;i<3;i++)
        cout << *(parray++) << endl; // σωστό
}
```

ΔΕΪΚΤΕΣ

Μεταβίβαση μεταβλητών με αναφορά

Παράδειγμα:

```
#include <iostream.h>
void centimize(double&)
void main()
{
    double var=2.3;
    cout << "var=" << var << " inches" << endl;
    centimize(var);
    cout << "var=" << var << " cm" << endl;
}

void centimize(double& v)
{
    v=v*2.54;
}
```

ΔΕΪΚΤΕΣ

Μεταβίβαση μεταβλητών με δείκτες

Παράδειγμα:

```
#include <iostream.h>
void centimize(double*)
void main()
{
    double var=2.3;
    cout << "var=" << var << " inches" << endl;
    centimize(&var);
    cout << "var=" << var << " cm" << endl;
}

void centimize(double* dpntr)
{
    *dpntr=(*dpntr)*2.54;
}
```

Μεταβίβαση πινάκων με δείκτες

```
#include <iostream.h>

const int MAX=10;

void centimize(double*, int);

void main()
{
    int N,i;
    double inchArray[MAX];
    cout << "Give the number of array elements: ";
    cin >> N;
    cout << "Give the array elements in inches:" << endl;
    for(i=0;i<N;i++)
        cin >> *(inchArray+i);
    centimize(inchArray,N);
    cout << "Array elements in cm:" << endl;
    for(i=0;i<N;i++)
        cout << *(inchArray+i) << endl;
};

void centimize(double* array, int arraySize)
{
    for(int i=0;i<arraySize;i++)
        *(array+i)=*(array+i)*2.54;
}
```

ΔΕΪΚΤΕΣ

Μεταβίβαση αλφαριθμητικών με δείκτες

Παράδειγμα:

```
#include <iostream.h>

void dispstr(char*);

void main( )
{
    char str[ ] = "I am a string";
    dispstr(str);
}

void dispstr(char* ps)
{
    while( *ps )
        cout << *ps++;
    cout << endl;
}
```

Διαχείριση μνήμης

- Δυναμική δέσμευση μνήμης
- Αποδέσμευση μνήμης
- Οι τελεστές `new` και `delete`

Διαχείριση μνήμης (1/4)

```
#include <iostream.h>

void centimize(double*, int);

void main()
{
    int N,i;
    double *inchArray;
    cout << "Give the number of array elements: ";
    cin >> N;
    inchArray=new double[N];
    cout << "Give the array elements in inches:" << endl;
    for(i=0;i<N;i++)
        cin >> *(inchArray+i);
    centimize(inchArray,N);
    cout << "Array elements in cm:" << endl;
    for(i=0;i<N;i++)
        cout << *(inchArray+i) << endl;
    delete inchArray;
};

void centimize(double* array, int arraySize)
{
    for(int i=0;i<arraySize;i++)
        *(array+i)=*(array+i)*2.54;
}
```

Διαχείριση μνήμης (2/4)

“String.h”

```
#include <iostream.h>
#include <string.h>
class String {
private:
    char* str;
public:
    String(char* s) {
        int length = strlen(s);
        str = new char[length+1];
        strcpy(str, s);
    }
    ~String( ) {
        delete str;
    }
    void display( ) {
        cout << str;
    }
};
```

“MainProg.cpp”

```
#include "String.h"
void main( )
{
    String s = "First";
    String* sp1=&s;
    String* sp2=
        new String("Second");

    cout << endl;
        s.display( );
    cout << endl ;
        sp1->display( );
    cout << endl ;
        sp2->display( );

    delete sp2;
}
```

Διαχείριση μνήμης (3/4)

“String.h”

```
#include <iostream.h>
#include <string.h>
class String {
private:
    char* str;
public:
    String(char* s) {
        int length = strlen(s);
        str = new char[length+1];
        strcpy(str, s);
    }
    ~String( ) {
        delete str;
    }
    void display( ) {
        cout << str;
    }
};
```

“MainProg.cpp”

```
#include "String.h"
void main( )
{
    String* s[2];
    s[1]=new String("First");
    s[2]=new String("Second");

    cout << endl;
        s[1]->display( );
    cout << endl ;
        s[2]->display( );

    delete s[1];
    delete s[2];
}
```

Διαχείριση μνήμης (4/4)

“String.h”

```
#include <iostream.h>
#include <string.h>
class String {
private:
    char* str;
public:
    String(char* s) {
        int length = strlen(s);
        str = new char[length+1];
        strcpy(str, s);
    }
    ~String( ) {
        delete str;
    }
    void display( ) {
        cout << str;
    }
};
```

“MainProg.cpp”

```
#include "String.h"
void main( )
{
    int i,nNames;
    char line[100];
    String** names;
    cout << "Number of names:";
    cin >> nNames;
    names=new String*[nNames];
    for(i=0;i<nNames;i++) {
        cin >> line;
        names[i]=new String(line);
    }
    cout << "The names are:"
        << endl;
    for(i=0;i<nNames;i++) {
        s[i]->display( );
        cout << endl;
    }
}
```

Αρχεία και ρεύματα

- Ρεύμα: ροή δεδομένων
- Διαφορετική ροή δεδομένων
⇒ Διαφορετικό ρεύμα
- Διαφορετικό ρεύμα
⇒ Διαφορετική τάξη

Αρχεία και ρεύματα

Αρχεία δίσκου



Είσοδος/Εξοδος αλφαριθμητικών

```
#include <fstream.h>
void main( )
{
    const LineSize=80;
    char buffer[LineSize];

    ofstream outfile("TEST.TXT");
    outfile << "Check this line\n";
    outfile.close();

    ifstream infile("TEST.TXT");
    while(infile)
    {
        infile.getline(buffer,LineSize);
        cout << buffer;
    }
    infile.close();
}
```

Είσοδος/Εξοδος αντικειμένων

“Person.h”

```
#include <iostream.h>
#include <string.h>
class Person {
private:
    char* name;
    int age;
public:
    Person(char* s, int a) {
        int length = strlen(s);
        name = new char[length+1];
        strcpy(name, s);
        age=a;
    }
    Person() { }
    ~Person( ) {
        delete name;
    }
    void show(void) {
        cout << name << ", " << age << endl;
    }
};
```

“MainProg.cpp”

```
#include "Person.h"
#include <fstream.h>
void main( )
{
    Person p1("Name",30),p2;
    ofstream outfile("TEST.DAT");
    outfile.write((char *)&p1, sizeof(p1));
    outfile.close();
    ifstream infile("TEST.DAT");
    infile.read((char *)&p2, size of(p2));
    infile.close();
    p2.show();
}
```


Υπέρβαση τελεστών εισαγωγής/εξαγωγής

“Person.h”

```
#include <iostream.h>
#include <string.h>
class Person {
private:
    char* name;
    int age;
public:
    Person(char* s, int a);
    Person() {}
    ~Person();
    friend ostream& operator <<
        (ostream& s, Person p);
}
```

“Person.cpp”

```
#include <iostream.h>
#include <string.h>
Person::Person(char* s, int a) {
    int length = strlen(s);
    name = new char[length+1];
    strcpy(name, s);
    age=a;
}
Person::~~Person( ) {
    delete name;
}

ostream& operator << (ostream& s,
    Person& p) {
    cout << name << ", " << age << endl;
    return s;
}
```

Άλλα θέματα

- Υπερβατικές συναρτήσεις
- Φίλες συναρτήσεις και τάξεις
- Συναρτήσεις `static`
- Εγκατάσταση με αντιγραφή
(και ο τελεστής `this`)